

Защита в операционных системах

Вязанкин Олег Николаевич

к.ф.-м.н., доцент кафедры ПОЗИ, ИМИТ

ovyazankin@gmail.com



Тема 3. Управление доступом.



План

- Основные определения
- Типовые модели управления доступом
- Управление доступом в UNIX
- Управление доступом в Windows

Основные определения

- *Объектом доступа* (или просто *объектом*) является любой элемент ОС, доступ к которому пользователей и других субъектов доступа может быть произвольно ограничен.

Ключевым словом в данном определении является слово **произвольно**. Если правила, ограничивающие доступ субъектов к некоторому элементу ОС, определены жестко и не допускают изменения с течением времени, этот элемент ОС мы не будем считать объектом доступа. Иначе говоря, возможность доступа к объектам ОС определяется текущей политикой безопасности, но не архитектурой ОС.

- *Методом доступа* к объекту называется операция, определенная для некоторого объекта.

Например, для семафора могут быть определены методы доступа “up” и “down”.

Основные определения

- *Субъектом доступа* (или просто *субъектом*) называется любая сущность, способная инициировать выполнение операций над объектами (обращаться к объектам по некоторым методам доступа).

Например, пользователи являются субъектами доступа. Обычно к субъектам доступа относят не только пользователей системы, но и порожденные ими процессы. Данный подход является оправданным и, более того, единственно верным во всех случаях, когда в область рассмотрения включаются программные закладки, функционирующие автономно и преследующие свои собственные цели, не совпадающие с целями пользователя, работающего в системе. Далее мы будем преимущественно рассматривать «чистые» ОС, поэтому везде далее, где явно не оговорено противное, субъектом доступа мы будем считать не процесс (или поток процесса-сервера), выполняющий некоторую операцию, а пользователя, от имени которого этот процесс (или поток) выполняется.

Основные определения

Итак, кратко:

- ❖ объект доступа – это то, к чему осуществляется доступ;
- ❖ субъект доступа – это тот, кто осуществляет доступ;
- ❖ метод доступа – это то, как осуществляется доступ.

■ Для объекта доступа может быть определен *владелец* – субъект, несущий ответственность за конфиденциальность содержащейся в объекте информации (если эта информация конфиденциальна), а также за целостность и доступность объекта.

Обычно владельцем объекта автоматически назначается субъект, создавший данный объект, в дальнейшем владелец объекта может быть изменен с использованием соответствующего метода доступа к объекту. Владелец объекта не может быть лишен некоторых прав на доступ к этому объекту, на владельца, как правило, возлагается ответственность за корректное ограничение прав доступа к данному объекту других субъектов.

Основные определения

- *Правом доступа* будем называть право на выполнение доступа к объекту по некоторому методу или группе методов доступа.
- Говорят, что *субъект имеет право на доступ к объекту* (или *субъект имеет право на объект*), если он имеет возможность осуществлять доступ к объекту по соответствующему методу или группе методов.

Например, если пользователь имеет возможность читать файл, говорят, что он имеет право на чтение этого файла.

Важно! Понятия *метода доступа* и *права доступа* не идентичны. Например, в ОС семейства UNIX право на запись в файл дает возможность субъекту обращаться к файлу как по методу «запись», так и по методу «добавление», при этом, поскольку право доступа «добавление» в UNIX отсутствует, невозможно разрешить субъекту операцию добавления, одновременно запретив операцию записи.

Основные определения

- Говорят, что субъект имеет некоторую *привилегию*, если он имеет возможность выполнять в ОС некоторые действия, не выражаемые или трудно выражаемые в терминах доступа субъекта к объектам.

Например, в ОС Windows поддерживаются привилегии перезагружать компьютер и перенастраивать часы компьютера.

- Как частный случай, *привилегией* является возможность применения некоторого права доступа или группы прав доступа ко всем без исключения объектам ОС, поддерживающим соответствующие методы доступа.

Например, если субъект ОС Windows имеет привилегию отладки, он имеет право доступа ко всем объектам типа «процесс» и «поток» по группе методов, используемых отладчиками при отладке программ (фактически, по всем поддерживаемым ОС методам доступа).

Основные определения

- Полномочиями субъекта доступа называется совокупность всех предоставленных ему прав и привилегий.
 - Управлением доступом субъектов к объектам называется совокупность правил, определяющая для каждой тройки субъект-объект-право, разрешена ли реализация данного права данным субъектом в отношении данного объекта.
- При дискреционном управлении доступом возможность доступа определяется для каждой тройки субъект-объект-право априорно, при мандатном управлении доступом ситуация несколько сложнее.
- Будем называть субъекта доступа *суперпользователем*, если он имеет возможность игнорировать правила управления доступом к объектам.

Основные определения

Правила управления доступом, действующие в защищаемой компьютерной системе (КС), устанавливаются администратором системы при определении текущей политики безопасности.

■ За соблюдением этих правил субъектами доступа следит *монитор ссылок* или *монитор безопасности объектов* – часть подсистемы защиты ОС.

Правила управления доступом должны удовлетворять следующим очевидным требованиям.

■ Правила управления доступом, принятые в КС, должны соответствовать аналогичным правилам, принятым в организации, в которой эксплуатируется данная система. Другими словами, если, согласно правилам организации, доступ пользователя к некоторой информации считается несанкционированным, то в ОС этот доступ тоже должен быть запрещен. Под НСД здесь подразумевается не только несанкционированное чтение информации, но и несанкционированное изменение, копирование или уничтожение информации.

Основные определения

- Правила управления доступом должны не допускать (или, по крайней мере, затруднять) разрушающие воздействия субъектов доступа, не обладающих соответствующими полномочиями, на операционную систему, выражающиеся в несанкционированном изменении, удалении или другом воздействии на объекты, критически важные для обеспечения нормального функционирования системы.
- Любой объект системы должен иметь владельца. Присутствие в системе *ничейных объектов* – объектов, не имеющих владельца, должно быть недопустимо.
- Присутствие в системе *недоступных объектов* – объектов, к которым не может обратиться ни один субъект доступа ни по одному методу доступа, должно быть недопустимо. Недоступные объекты фактически бесполезно растрачивают аппаратные и программные ресурсы КС.
- Утечка конфиденциальной информации из защищаемой системы должна быть недопустима. Поскольку реализовать выполнение данного требования программно-аппаратными средствами весьма сложно, оно предъявляется лишь в редких случаях. Как правило, это требование обеспечивается одними только организационными мерами.



Типовые модели управления доступом

- Дискреционное управление доступом.
- Изолированная программная среда.
- Мандатное управление доступом.

Управление доступом в UNIX

Объекты доступа в UNIX:

- файлы,
- директории,
- ссылки (links),
- устройства,
- именованные каналы (named pipes)
- процессы (/proc).

Субъекты доступа:

- пользователи (включая псевдопользователей),
- группы пользователей,
- root.

Пользователи и группы пользователей идентифицируются числовыми идентификаторами (UID – User ID, GID – Group ID). UID для root'a равен 0.

```
instructor@ubuntu:~$ id
uid=1001(instructor) gid=1001(instructor) группы=4(adm),
20(dialout),21(fax),24(cdrom),25(floppy),26(tape),30(dip)
,44(video),46(plugdev),104(fuse), 1001(instructor)
```

Управление доступом в UNIX

Роль *действительного* (работающего с объектами) субъекта играет *процесс*. Каждый процесс снабжен единственным UID: это идентификатор запустившего процесс номинального субъекта, т. е. пользователя.

Процесс, порожденный некоторым процессом пользователя, наследует его UID (из этого правила есть некоторые исключения, обеспечивающие штатные для системы способы повышения и понижения уровня доступа процесса). Таким образом, все процессы, запускаемые по желанию пользователя, будут иметь его идентификатор.

UID учитываются, например, когда один процесс посылает другому сигнал. В общем случае разрешается посылать сигналы «своим» процессам (тем, что имеют такой же UID).

Управление доступом в UNIX

Права доступа

Роль основного *объекта* доступа в UNIX играют объекты файловой системы. В архитектуре UNIX в файловой системе представлены не только обычные файлы с данными, там присутствуют и специальные файлы для устройств, каналов, сокетов и т. д. Благодаря этому регулирование доступа к файлам позволяет покрыть очень широкий спектр ситуаций доступа и служит основным средством организации политики доступа в UNIX.

В соответствии с субъект-субъектной моделью каждый файл снабжен ярлыком, в котором хранятся идентификаторы номинальных субъектов, которые вправе распоряжаться доступом к данному файлу. В случае UNIX — это идентификатор пользователя-владельца (UID) и идентификатор группы-владельца (GID). **Обратите внимание**, что файл может принадлежать только одной группе, в то время как пользователь может входить в несколько групп.

Управление доступом в UNIX

Права доступа

На уровне файловой системы в UNIX определяются *три вида доступа*: чтение (**read**, *r*), запись (**write**, *w*) и использование (**execution**, *x*). Право на чтение из файла дает доступ к содержащейся в нем информации, а право записи — возможность ее изменять. Право *использования (исполнения)* имеет смысл не для всех типов файлов, хотя может быть установлено для любого.

В случае обычного файла оно означает возможность *исполнения* файла, т.е. запуска программы или командного сценария, содержащихся в этом файле. Право на *исполнение* в случае каталога означает, что в него можно заходить (*cd*), просматривать и изменять содержимое файлов, если на них есть разрешения, узнавать свойства файлов (метаданные).

Для устройств каждое право доступа дает возможность обращения к соответствующей функции драйвера, обслуживающего данное устройство.

Для именованного канала право **read** требуется для получения информации, право **write** — для отправки информации, право **execution** не определено.

Управление доступом в UNIX

Права доступа

Выделяют три категории пользователей, которым могут предоставляться права на файл:

- Сам владелец (**u – user**) объекта – конкретный пользователь, чье имя числится в атрибутах файла как имя владельца этого файла. Обычно если пользователь создает файл, то он автоматически записывается как его владелец.
- Группа (**g – group**), к которой принадлежит владелец файла. Когда в UNIX создается пользователь, то для него создается одноименная группа. Однако средствами администрирования системы можно объединять пользователей в различные группы. При этом конкретный пользователь может входить в состав нескольких групп. Группы позволяют предоставлять права доступа к объектам сразу нескольким людям, но при этом ограниченному кругу лиц.
- Все остальные (**o – other**) – это все те, кто не является владельцем файла и не принадлежит к группе владельца файла. То есть любой другой пользователь.

Управление доступом в UNIX

Права доступа

При каждом файле имеется ярлык, в котором зафиксированы права доступа к нему. Права доступа включают список из девяти пунктов (три тройки): по три вида доступа для трех групп — пользователя-владельца, группы-владельца и всех остальных. Каждый пункт в этом списке может быть либо разрешён, либо запрещён (равен 0 либо 1). Таким образом, для хранения этой информации о правах доступа достаточно 9 бит.

The diagram shows a line from a UNIX file listing: `-rw-r--r-- 1 tokza wheel 8480 Nov 14 00:47 file.txt`. Below this line, seven red arrows point upwards to specific parts of the line, each arrow originating from a black circle containing a white number from 1 to 7. The arrows point to: 1. the permission string, 2. the number of hard links, 3. the owner name, 4. the group name, 5. the file size, 6. the date and time, and 7. the filename.

1. права доступа (владельца, группы-владельца, остальных)
2. количество жестких ссылок
3. владелец файла
4. группа-владелец файла
5. размер файла в байтах
6. дата последней модификации файла
7. имя файла

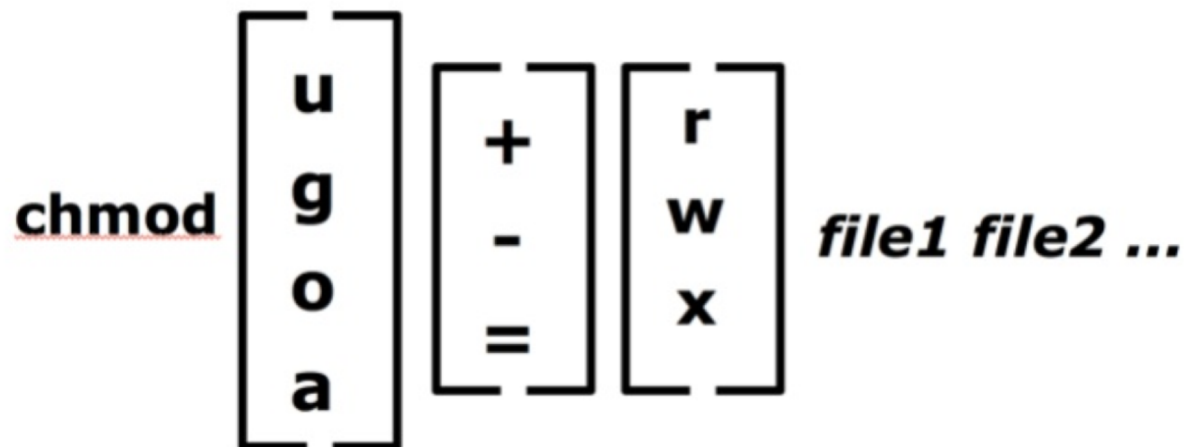
Типы файлов:

- - - file
- d – dir
- l – link
- c – char dev
- b – block dev
- s – socket
- p - FIFO

Управление доступом в UNIX

Права доступа

Установка битов доступа



u — user

g — group

o — other

a - all

+ —добавить

- — убрать

= — установить

```
chmod a+w file1
```

```
chmod g-rw file2
```

```
chmod go=r file3
```

Управление доступом в UNIX

Права доступа

Установка битов доступа

- | 1 — бит доступа установлен
- | 0 — бит доступа не установлен

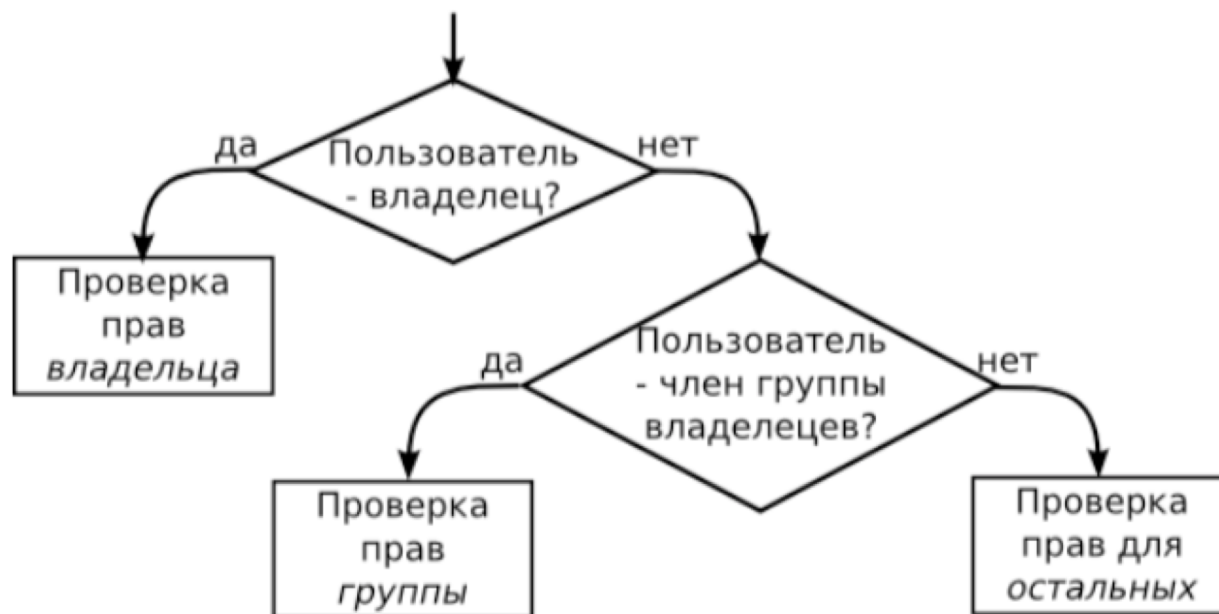
-rwx r-x r-- → 111 101 100 → 754

```
# chmod 754 somefile
```

Управление доступом в UNIX

Права доступа

При обращении процесса к файлу (с запросом доступа определённого вида, т.е. на чтение, запись или исполнение) система проверяет совпадение идентификаторов владельцев процесса и владельцев файла в определённом порядке, и в зависимости от результата, применяет ту или иную группу прав.





Управление доступом в UNIX

Права доступа

Строго говоря, при этом проверяется не собственно идентификатор пользователя процесса (UID), а т.н. *исполнительный идентификатор пользователя*, EUID. Он существует в связи с тем, что в ходе выполнения процесс может менять субъект, от имени которого он выполняется. Подробнее об этом сказано в разделе **Подмена идентификатора процесса**.

Управление доступом в UNIX

Разделяемые каталоги

Право на запись для каталога трактуется как возможность создания и удаления файлов в нём, а также возможность изменения атрибутов файлов (например, переименование). При этом субъекту не обязательно иметь права на запись для этих файлов, поскольку и переименование и удаление файла затрагивают только сам каталог.

Таким образом, из своего каталога пользователь может удалить любой файл. Часто возникает ситуация, когда каталог нужно использовать совместно — в этом случае необходимо разрешить запись в него либо группе пользователей, либо всем пользователям (например, общесистемный каталог для временных файлов /tmp). А если запись в каталог разрешена всем, то любой пользователь сможет удалить в нём любой файл. Для избежания этой проблемы был добавлен специальный атрибут— *sticky bit*. При установке этого атрибута пользователь, имеющий доступ на запись в этот каталог, может изменять только **принадлежащие ему** файлы.

Управление доступом в UNIX

Подмена идентификатора процесса

Каждому процессу UNIX присваиваются четыре числовые идентификатора:

- UID – идентификатор пользователя, породившего данный процесс;
- GID – идентификатор группы пользователя, породившего данный процесс;
- EUID – эффективный идентификатор пользователя;
- EGID – эффективный идентификатор группы пользователя.

Обычно EUID совпадает с UID, а EGID – с GID. Однако при подмене идентификатора процесса это не так.

Управление доступом в UNIX

Подмена идентификатора процесса

В UNIX существует механизм *подмены идентификатора* (SetUID, SUID), позволяющий пользователям запускать процессы с идентификаторами других пользователей. Этот механизм применяется в тех случаях, когда процессу для выполнения определённых операций необходимо предоставить повышенные права (например, суперпользователя) или права другого пользователя.

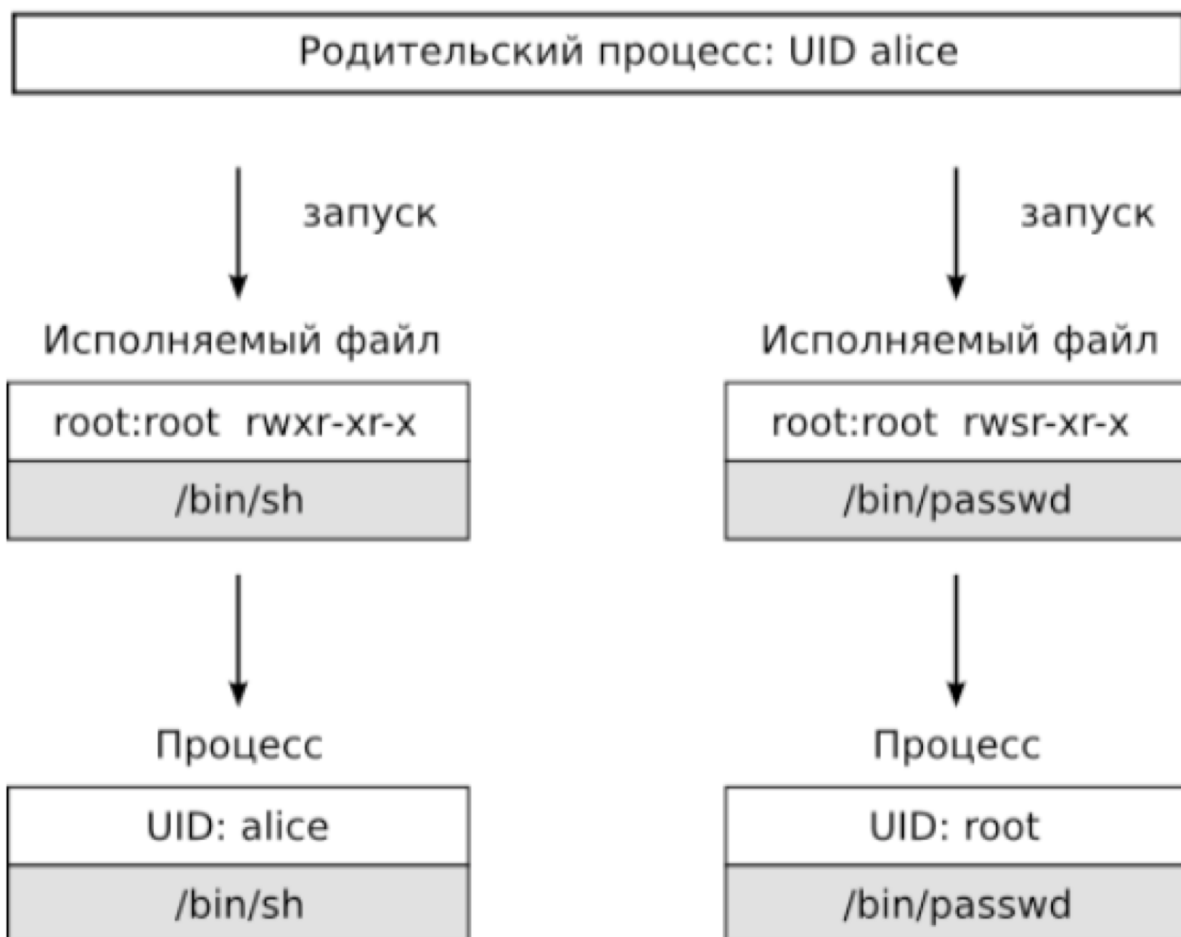
Подмена идентификатора происходит в том случае, если процесс запустит вместо себя при помощи системного вызова `exec()` программу *из файла*, в правах доступа которого установлен *бит подмены идентификатора пользователя* (SUID-бит, обозначается *s* (*S*) в символьной записи прав доступа). Запущенный из этого файла процесс получит исполнительный идентификатор владельца (EUID) файла вместо идентификатора владельца процесса-родителя (см. рисунок), благодаря чему UID процесса сохраняет информацию о том, кто *на самом деле* запустил программу.

Управление доступом в UNIX

Подмена идентификатора процесса

Запуск процесса:

Запуск suid-процесса:



Управление доступом в UNIX

Подмена идентификатора процесса

В современных UNIX-системах предусмотрен и ещё один дополнительный атрибут — SetGID (SGID), бит подмены идентификатора группы. Этот механизм работает совершенно аналогично подмене идентификатора пользователя, с тем отличием, что процесс, запущенный из файла с атрибутом SetGID, получает идентификатор группы-владельца файла, UID же его остается неизменным. Использование SetGID позволяет весьма гибким образом контролировать ситуацию повышения прав доступа процесса.

Особое значение имеют атрибуты подмены идентификатора (SetUID и SetGID), установленные на каталогах — для каталогов тоже используются права на исполнение, хотя и имеют другой смысл, чем у исполняемых файлов. Атрибут SetGID, установленный на каталоге, указывает, что файлы и подкаталоги, создаваемые внутри этого подкаталога любыми процессами, будут получать тот же идентификатор группы, что и сам каталог. Причем подкаталоги будут также наследовать атрибут SetGID. Такой механизм используется для организации общих каталогов, файлы в которых должны быть доступны на равных условиях группе пользователей. Атрибут SetUID, установленный на каталоге, просто игнорируется.

Атрибуты подмены идентификаторов пользователя и группы несут потенциальную угрозу безопасности системы и должны использоваться с осторожностью.

Управление доступом в UNIX

Установка дополнительных битов доступа

Цифровой формат записи

4000 — SUID

2000 — SGID

1000 — Sticky

```
chmod 4755 file2
```

```
chmod 2750 file3
```

```
chmod 1777 dir2/
```

Управление доступом в UNIX

Права доступа по умолчанию

Процесс может создавать объект с любыми атрибутами. В общем случае объект создается со следующими атрибутами безопасности:

- владелец (UID) – EUID процесса;
- группа (GID) – EGID процесса, либо GID каталога, в котором создается объект (в BSD-системах);
- права доступа – согласно `umask`.

При создании файла или директории, среда операционной системы присваивает им определенные права доступа по умолчанию, и `umask` - это пользовательская маска (**user mask**), которая используется для определения конечных прав доступа.

Текущую маску можно узнать, набрав в командной строке `umask` без параметров. Маска по умолчанию:

- 0002 – для обычных пользователей;
- 0022 – для root'a

Управление доступом в UNIX

Права доступа по умолчанию

В UNIX существуют базовые права:

- для файла - 0666 (rw-rw-rw-);
- для директории - 0777 (rwxrwxrwx).

При создании объекта на эти базовые права накладывается маска. Она указывает, какие биты следует сбросить в выставляемых правах на файл — каждый установленный бит `umask` запрещает выставление соответствующего бита прав. Исключением из этого запрета является бит исполняемости, который для обычных файлов зависит от создающей программы (трансляторы ставят бит исполняемости на создаваемые файлы, другие программы — нет), а для каталогов следует общему правилу.

Например, для маски 0002 права по умолчанию для директории равны 775 (rwxrwxr-x), а для файла - 664 (rw-rw-r--).

Исключения:

- Флаг “-p” у команды `cp` сохраняет права доступа при копировании. SUID/SGID флаги сбрасываются, если команду запустил не `root`.
- При перемещении файла (`mv`) сохраняются все биты доступа, но для этого надо иметь бит записи в правах на оба каталога.

Управление доступом в UNIX

Суперпользователь

Пользователь root (он же *суперпользователь*) имеет нулевые UID и GID и играет роль *доверенного субъекта UNIX*. Это значит, что он не подчиняется законам, которые управляют правами доступа, и может по своему усмотрению эти права изменять. Большая часть общезначимых (не принадлежащих конкретным пользователям) компонентов системы доступна для модификации только суперпользователю.

Суперпользователь работает на уровне доступа ядра, так что он является, по сути, неотъемлемым компонентом самой системы.

Многие команды должны исполняться только от имени суперпользователя, так как в них производится взаимодействие с частями ядра, отвечающими за взаимодействие с аппаратурой, правом доступа и т. п. Если же такие команды разрешается запускать простым пользователям, применяется рассмотренный выше механизм подмены идентификатора пользователя.

Управление доступом в UNIX

Суперпользователь

Администрирование в UNIX (т. е. управление общезначимыми характеристиками системы) требует привилегий суперпользователя. При работе с повышенными привилегиями, в особенности от имени пользователя *root* следует быть очень осторожным: выполнение неверной команды может привести к выходу системы из строя и утрате информации.

Если, например, в команде `rm -rf /*.bak` ошибочно поставить пробел между звездочкой и точкой, сама ОС и все хранящиеся в ней данные будут немедленно и необратимо уничтожены.

Поэтому даже администраторы никогда не работают в командной оболочке с правами суперпользователя всё время, а переходят в режим суперпользователя только тогда, когда это действительно необходимо (например, с помощью команд `su` (**S**ubstitute **U**ser) и `sudo` (**S**ubstitute **U**ser **DO**)).

Управление доступом в UNIX

Классическая модель управления доступом DAC. Итоги

Каждый объект UNIX имеет атрибуты защиты, включающие в себя три элемента:

- идентификатор владельца объекта (UID);
- идентификатор группы-владельца объекта (GID);
- вектор доступа.

Вектор доступа включает в себя следующие элементы:

- тип объекта – 1 бит;
- бит SUID;
- бит SGID;
- бит sticky;
- права доступа владельца – 3 бита (rwx);
- права доступа пользователей, входящих в группу владельца – 3 бита (rwx);
- права доступа всех остальных пользователей – 3 бита (rwx).

Управление доступом в UNIX

Ограничения базовой модели управления доступом и ее расширения

Простота системы прав доступа UNIX приводит к некоторым ограничениям.

- Ограниченный набор прав доступа (rwx).
- Нет возможности дать разным группам пользователей разные права доступа.
- Управлять группами может только root.
- Пользователь имеет контроль над своими объектами и может снизить их безопасность.

Решения:

- дополнительные права доступа;
- POSIX ACL;
- POSIX capabilities (возможности, привилегии);
- альтернативные модели безопасности (MAC, RBAC, DTE).

Управление доступом в UNIX

Дополнительные права доступа

В UNIX безопасность устроена очень просто по принципу «все (root), или ничего (почти ничего для всех остальных пользователей)». Возникают ситуации, когда необходимо защититься от всемогущего root'a.

Появились дополнительные права доступа, ограничивающие возможности стандартных операций над объектами (файлами).

- **append_only**: для файла – только дописывать (если есть право w), для каталога – только создавать в нем файлы.
- **nounlink**: запрещает удалять или переименовывать файл/каталог.
- **immutable**: для файла – ничего нельзя, кроме чтения/копирования, для каталога – нельзя создавать и удалять файлы, а также переименовывать каталог.

Управление:

- Linux: `chattr`;
- BSD: `chflags`.

Управление доступом в UNIX

Дополнительные права доступа

Есть команды управления дополнительными правами доступа, но кто может ими пользоваться?

Система может находиться в доверенном (single user mode) и в не доверенном состоянии.

Например, в FreeBSD есть концепция **securelevel** (-1, 0, 1, 2, 3). При загрузке системы **securelevel** стартует с уровня -1 и пока уровень не повысится до 2 (система все еще находится в доверенном состоянии) можно управлять дополнительными правами доступа.

Управление доступом в UNIX

POSIX ACL

- Дает возможность гранулированно выставить права на объекты для каждого пользователя/группы в системе.
- Требуется поддержки от файловой системы.
- *ACL дополняет обычные права доступа, а не замещает их.*

```
$ setfacl -m group:mygroup:rw- myfile1
$ setfacl -m user:bob:rwx myfile2
$ getfacl myfile2
# file: myfile2
# owner: root
# group: root
user::rw-
user:bob:rwx
group::r-
group:mygroup:rw-
mask::rw-
other::r--
```

Управление доступом в UNIX

POSIX ACL

При использовании *ACL* надо держать в уме, что при некоторых файловых операциях можно потерять эти настройки. Например:

- Копировать файл с *ACL* необходимо с опцией `cp -p file.txt ~/`, но `mv file.txt ~/` и без дополнительных опций сохранит *ACL*.
- При копировании файла с тома, смонтированного с опцией `acl`, на том `noacl`, установленные *ACL* потеряются.
- Потерять *ACL* можно при бэкапе средствами, не умеющими работать с *ACL*. Для обхода этих ограничений можно сохранять настройки *ACL* в отдельный файл с помощью `getfacl`, с последующим `setfacl` из файла при операции восстановления их архива.
- Не все утилиты командной строки поддерживают *ACL*.

Управление доступом в UNIX

POSIX capabilities

Понятие привилегии субъекта доступа в ранних версиях UNIX отсутствовало. В дальнейшем эта концепция вводилась в разных ветвях эволюции UNIX независимо друг от друга. В наиболее распространенной на сегодняшний день ее реализации, применяемой в частности Linux, привилегии называются *возможностями (capabilities)*.

- **CAP_CHOWN** – позволяет модифицировать идентификаторы владельца или группы владельца любого объекта;
- **CAP_SETUID** – позволяет устанавливать бит SUID в векторе доступа объекта;
- **CAP_SETGID** – позволяет устанавливать бит SGID в векторе доступа объекта;
- **CAP_LINUX_IMMUTABLE** - позволяет установку дополнительных атрибутов EXT2_APPEND_FL и EXT2_IMMUTABLE_FL для файловой системы ext2;
- **CAP_NET_BIND_SERVICE** – позволяет привязку к зарезервированным портам сокетов доменов интернет (то есть к номерам портов менее 1024);
- **CAP_SETPCAP** - разрешает или запрещает любые возможности в допустимых возможностях вызывающего процесса, установленных в/из любых других процессах.

Управление доступом в Windows

Объекты доступа

- В ОС семейства MS Windows **все** объекты ОС являются объектами доступа. Иначе говоря, доступ субъектов к любому объекту ОС может быть произвольно ограничен.
- Атрибуты защиты объекта Windows входят в число обязательных атрибутов, без них объект физически не может существовать. Даже те объекты, которые не могут иметь атрибутов защиты (например, файл в FAT), при открытии получают временный набор атрибутов защиты «объект общедоступен».
- В Windows набор типов объектов не задан жестко в коде ОС, он может расширяться системным ПО, в том числе и разработанным третьими фирмами.

В Windows 7 и выше определено более 40 стандартных типов объектов, большинство из которых используются внутри ОС и недоступны прикладным программам.

Управление доступом в Windows

Объекты доступа

Как правило, прикладные программы работают только с объектами следующих типов:

- файловые объекты (файлы, каталоги, устройства, именованные каналы и т.п.);
- ключи реестра;
- секции разделяемой памяти;
- процессы и потоки;
- мьютексы;
- семафоры;
- порты;
- маркеры доступа;
- рабочие столы;
- оконные станции;
- пакетные задания;
- директории дерева объектов (\ObjectTypes);
- символические связи (линки) дерева объектов.

Управление доступом в Windows

Субъекты доступа

ОС Windows поддерживает следующие типы субъектов доступа:

- *Пользователи, включая псевдопользователей.* К последним относятся следующие субъекты доступа:
 - SYSTEM – ОС локального компьютера. Данный пользователь всегда входит в группу Administrators и всегда имеет все привилегии;
 - LOCAL SERVICE – псевдопользователь, от имени которого выполняются сетевые сервисы;
 - ANONYMOUS - «бесправный» псевдопользователь, от имени которого выполняются сетевые запросы, сделанные в рамках нуль-сессии (null session);
 - <<имя_компьютера>>\$ - псевдопользователи, соответствующие компьютерам, входящим в домен. Они используются при взаимной аутентификации компьютеров в лесу доменов и для делегирования полномочий псевдопользователю SYSTEM одного компьютера на другие компьютеры леса доменов.
- *Группы пользователей.* Пользователь может входить в потенциально неограниченное количество групп. Среди всех групп, в которые входит пользователь, выделяется одна *первичная группа* (для совместимости со стандартом POSIX) .

Управление доступом в Windows

Субъекты доступа

- *Специальные временные группы.* В отличие от обычных групп членство пользователя в таких группах определяется не администратором, а самой ОС в зависимости от способа взаимодействия пользователя с системой. К специальным группам относятся:
 - INTERACTIVE – пользователи, работающие с системой локально (обычно не более одного);
 - NETWORK – пользователи, работающие с системой через сеть;
 - DIAL_UP – пользователи, работающие с системой по модему;
 - BATCH – пользователи и псевдопользователи, от имени которых запущены пакетные задания (batch jobs);
 - SERVICE – пользователи и псевдопользователи, от имени которых выполняются сервисы;
 - TERMINAL SERVER USER – пользователи, работающие с системой через терминальную сессию.

Управление доступом в Windows

Субъекты доступа

- *Относительные субъекты.* Эти субъекты определяются относительно объекта, для которого определяются права доступа. Существуют следующие относительные субъекты:
 - CREATOR_OWNER – владелец объекта;
 - CREATOR_GROUP – первичная группа владельца объекта.

Относительные субъекты используются, если нужно описать права доступа пользователей к объектам по принципу «что кому принадлежит, то ему и доступно».

Управление доступом в Windows

Идентификатор безопасности SID. Структура идентификатора безопасности

- Для идентификации субъектов доступа в Windows используется особый тип идентификатора, называемый SID (Security ID).
- Предопределенные субъекты доступа, перечисленные выше, а также группа Everyone (группа, в которую входят все пользователи, возможно, за исключением псевдопользователя ANONYMOUS) имеют стандартные идентификаторы, общие для всех экземпляров ОС.
- Для всех остальных пользователей идентификаторы безопасности генерируются при создании учетной записи и они уникальны в пределах всей вселенной (исключение – некорректное «клонирование» ОС из одной резервной копии).

Именно SID (а не имена пользователей, которые могут не быть уникальными) служат основой для идентификации субъектов внутренними процессами ОС Windows.



Управление доступом в Windows

Идентификатор безопасности SID. Структура идентификатора безопасности

Идентификатор безопасности хранится в бинарном виде в базе данных менеджера учетных записей SAM (Security Account Manager).

Существует также и текстовая форма представления SID. Текстовая форма используется для вывода текущего значения SID, а также для интерактивного ввода (например, в реестр).

В текстовой форме каждый идентификатор безопасности имеет определенный формат. Вначале находится префикс S, за которым следует группа чисел, разделенных дефисами.

Управление доступом в Windows

Идентификатор безопасности SID. Структура идентификатора безопасности

Символически структура идентификатора безопасности может быть описана следующим образом:

$$S - R - I - SA_0 - SA_1 - SA_2 - SA_3 - SA_4 \dots$$

Здесь каждый символ обозначает группу бит, имеющих определенное значение, а именно:

- S – представляет символ S, который обозначает, что дальнейшее числовое значение является идентификатором безопасности;
- R – представляет версию (RevisionLevel) формата идентификатора безопасности. Начиная с ОС Windows NT версии 3.1, формат идентификатора безопасности не изменялся и поэтому значение R всегда равно 1;
- I – представляет 48-битное число, которое обозначает *уровень авторизации учетной записи* (Toplevel Authority или Identifier Authority), которая связана с данным идентификатором безопасности. Это значение также называется *идентификатором авторизации учетной записи*. Под уровнем авторизации понимается уровень, на котором была создана учетная запись;

Управление доступом в Windows

Идентификатор безопасности SID. Структура идентификатора безопасности

- SA_i – представляет 32-битное число, которое уточняет уровень авторизации учетной записи (Subauthority), связанной с данным идентификатором безопасности. Это число также называется *относительным идентификатором учетной записи* (Relative Identifier, RID). Относительные идентификаторы учетной записи предназначены для конкретизации или, другими словами, однозначной идентификации учетных записей. В общем случае количество битовых полей типа SA в идентификаторе безопасности может быть произвольным.

Для примера, SID администратора системы имеет вид: S–1–5–<домен>–500, а SID группы Everyone, в которую входят все пользователи, исключая анонимных: S–1–1–0.

Фактически идентификатор безопасности идентифицирует пользователя на уровне системы безопасности. Использование идентификатора безопасности ускоряет работу системы безопасности, т.к. в этом случае система при идентификации пользователей работает с числовыми, а не символьными данными.

Управление доступом в Windows

Методы и права доступа

ОС Windows поддерживает до 22 методов доступа субъектов к объектам каждого типа. Шесть методов доступа представляют собой *стандартные методы*, поддерживаемые для объектов всех типов:

- удаление объекта;
- получение атрибутов защиты объекта;
- изменение списка доступа объекта;
- изменение владельца объекта;
- получение и изменение параметров аудита в отношении объекта;
- ожидание объекта.

Для каждого типа объекта поддерживается до 16 *специфичных методов доступа* (см, например, Проскурин В.Г., 2014). Некоторые из этих методов доступа требуют наличия у субъекта доступа специальных привилегий.

Управление доступом в Windows

Методы и права доступа

Каждому специфичному методу доступа, поддерживаемому в Windows, соответствует *право* на его осуществление. Эти права доступа также называются *специфичными*, поскольку они специфичны для каждого типа объектов.

Каждому стандартному методу доступа, за исключением метода «получение и изменение параметров аудита в отношении объекта», также соответствует право доступа, дающее возможность реализации соответствующего метода доступа. Такие права доступа называются *стандартными*.

Для некоторых типов объектов стандартные и специфичные права доступа реализованы не вполне корректно (например, при попытке получения атрибутов защиты объекта «процесс» проверяется не стандартное право «получение атрибутов защиты», а специфичное право «получение информации о процессе»).

Управление доступом в Windows

Методы и права доступа

Также Windows поддерживает так называемые *общие* (*generic*) или *отображаемые* (*mapped*) права доступа к объекту любого типа:

- чтение (GENERIC_READ);
- запись (GENERIC_WRITE);
- выполнение (GENERIC_EXECUTE);
- все действия (GENERIC_ALL).

Каждое из отображаемых прав доступа представляет собой некоторый набор стандартных и специфичных прав доступа, зависящий от типа объекта доступа. Сделано это для того, чтобы облегчить настройку прав доступа для разработчиков и администраторов системы.

Следует иметь в виду, что порядок отображения общих прав доступа в набор стандартных и специфичных прав не обязательно совпадает с интуитивным смыслом общего права доступа.

Управление доступом в Windows

Методы и права доступа

Последним классом прав доступа, поддерживаемых Windows, являются *виртуальные права доступа*. Виртуальные права доступа **не могут быть предоставлены субъекту, но могут быть им запрошены**.

Поддерживаются два виртуальных права доступа:

- `MAXIMUM_ALLOWED`;
- `ACCESS_SYSTEM_SECURITY`.

Запрашивая виртуальное право `MAXIMUM_ALLOWED` на доступ к объекту, субъект тем самым требует открытия объекта с максимально доступными ему правами. ОС при этом проводит детальный анализ прав доступа данного субъекта по отношению к данному объекту.

Виртуальное право `ACCESS_SYSTEM_SECURITY` – это право на получение и изменение параметров аудита по данному объекту. При этом субъект должен обладать соответствующей привилегией доступа и это будет означать возможность реализовать данное право субъекта к любому объекту ОС. Поэтому данное право доступа является виртуальным.

Управление доступом в Windows

Методы и права доступа

Итак, в ОС Windows имеются следующие классы прав доступа и отвечающие им методы доступа:

- стандартные права;
- специфичные права;
- общие (отображаемые) права;
- виртуальные права.

Управление доступом в Windows

Привилегии субъектов доступа

Каждый пользователь и псевдопользователь Windows обладает некоторым (возможно, пустым) набором привилегий. Привилегии представляют собой права на выполнение субъектом действий, касающихся всей системы в целом, а не отдельных ее субъектов.

Перечислим некоторые основные привилегии, поддерживаемые в современных версиях Windows:

- создавать резервные копии информации, хранящейся на жестких дисках;
- восстанавливать информацию на жестких дисках с резервных копий;
- отлаживать программы;
- загружать и выгружать драйверы и сервисы;
- объявлять себя владельцем любого объекта;
- добавлять записи в журнал аудита.

Администраторы ОС должны подходить к назначению пользователям привилегий с максимальной ответственностью. Особое внимание следует уделять вышеперечисленным опасным привилегиям.



Управление доступом в Windows

Привилегии субъектов доступа

Помимо явно определенных привилегий, в Windows существуют также привилегии, неявно определенные через предопределенные группы.

Например, группа Administrators обладает целым рядом неявно определенных привилегий, например, привилегией регистрировать любых пользователей, привилегией переназначать владельца любого объекта и т.д.

Управление доступом в Windows

Маркер доступа

Субъекты, как впрочем и объекты, должны иметь отличительные признаки – контекст субъекта, для того чтобы система могла контролировать их действия. При интерактивном входе в систему пользователь обычно вводит свое имя и пароль. Система (процедура winlogon) по имени находит соответствующую учетную запись, извлекает из нее необходимую информацию о пользователе, формирует список привилегий, ассоциированных с пользователем и его группами, и все это объединяет в структуру данных, которая называется *маркером доступа* (*access token*). Этот маркер присваивается начальному процессу пользователя (как правило, Windows Explorer). Маркер доступа имеют также и псевдопользователи.

Каждый процесс, порожденный в дальнейшем пользователем, получает свою копию маркера доступа пользователя, эта копия является обязательным атрибутом процесса. Процесс, не имеющий маркера доступа, не может существовать. Также маркеры доступа могут назначаться отдельным потокам.

Управление доступом в Windows

Маркер доступа

Заголовок	Срок действия	Группы	DACL	Ограниченные идентификаторы SID	SID пользователя	SID группы	Привилегии
-----------	---------------	--------	------	---------------------------------	------------------	------------	------------

- Поле «Заголовок» содержит некоторую административную информацию.
- Поле «Срок действия» может сказать, когда маркер утрачивает актуальность (в настоящее время не используется).
- Поле «Группы» содержит список SID групп, которым принадлежит субъект.
- Поле «DACL» (Discretionary ACL) – это список управления доступом, присваиваемый создаваемым объектам по умолчанию (если не указан другой ACL).
- Поле «SID пользователя» говорит о том, кто владеет процессом.
- Ограниченные идентификаторы SID позволяют ненадежным процессам принимать участие в заданиях вместе с надежными процессами (и при этом у них меньше возможностей что-то испортить).
- Наконец, привилегии (если они есть) дают процессу специальные возможности (которых нет у обычных пользователей).

Управление доступом в Windows

Олицетворение

Когда пользователь регистрируется, то winlogon дает начальному процессу маркер доступа. Следующие процессы обычно наследуют этот маркер. Маркер доступа процесса первоначально применяется ко всем потокам процесса. Однако поток при выполнении может получить другой маркер доступа (так называемый *маркер олицетворения* – *impersonation access token*), в этом случае маркер доступа потока замещает маркер доступа процесса.

Механизм олицетворения обычно используется процессами-серверами. Когда процесс-сервер обслуживает запрос процесса-клиента, для выполнения запроса внутри процесса-сервера создается поток. Этот поток может получить права доступа от клиентского потока, чтобы сервер мог обратиться к защищенным файлам (и прочим объектам) клиента.

Механизм олицетворения реализован в транспортных слоях (например, ALPC, именованные каналы, TCP/ IP), используемых в RPC для обмена между клиентами и серверами. Транспортные слои используют внутренние интерфейсы монитора безопасности ядра, извлекая контекст безопасности для маркера доступа текущего потока и отправляя его на сервер, где он используется для конструирования маркера, который сервер может использовать для олицетворения клиента.

Управление доступом в Windows

Олицетворение

Если бы олицетворение клиентов не поддерживалось, при каждом обращении процесса-клиента к объекту ОС сервера процессу-серверу приходилось бы явно проверять достаточность прав доступа процесса-клиента к данному объекту сервера.

При этом достаточно пропустить всего лишь одну проверку прав доступа клиента и в системе появляется критическая уязвимость. Но если в ОС поддерживается механизм олицетворения, процесс-сервер может не заботиться о полномочиях клиента, все необходимые проверки делаются автоматически.

В ранних версиях Windows (до версии XP SP1 включительно) возможность олицетворять клиентов предоставлялась всем субъектам ОС без всяких ограничений. Это было серьезной потенциальной уязвимостью, легко приводящей к реальным уязвимостям, чаще всего критическим с точки зрения безопасности.

Управление доступом в Windows

Олицетворение

Например, если низко привилегированному пользователю удалось зарегистрировать свой процесс как сервер многопользовательского интерфейса и заставить высоко привилегированного пользователя подключиться к этому серверу, первый из них мог легко повысить свои полномочия, проведя олицетворение высоко привилегированного пользователя-клиента.

Начиная с Windows XP SP2, олицетворение клиента требует наличия у пользователя (псевдопользователя), от имени которого выполняется процесс-сервер, специальной привилегии «олицетворение клиентов».

Управление доступом в Windows

Дескриптор защиты объекта (дескриптор безопасности)

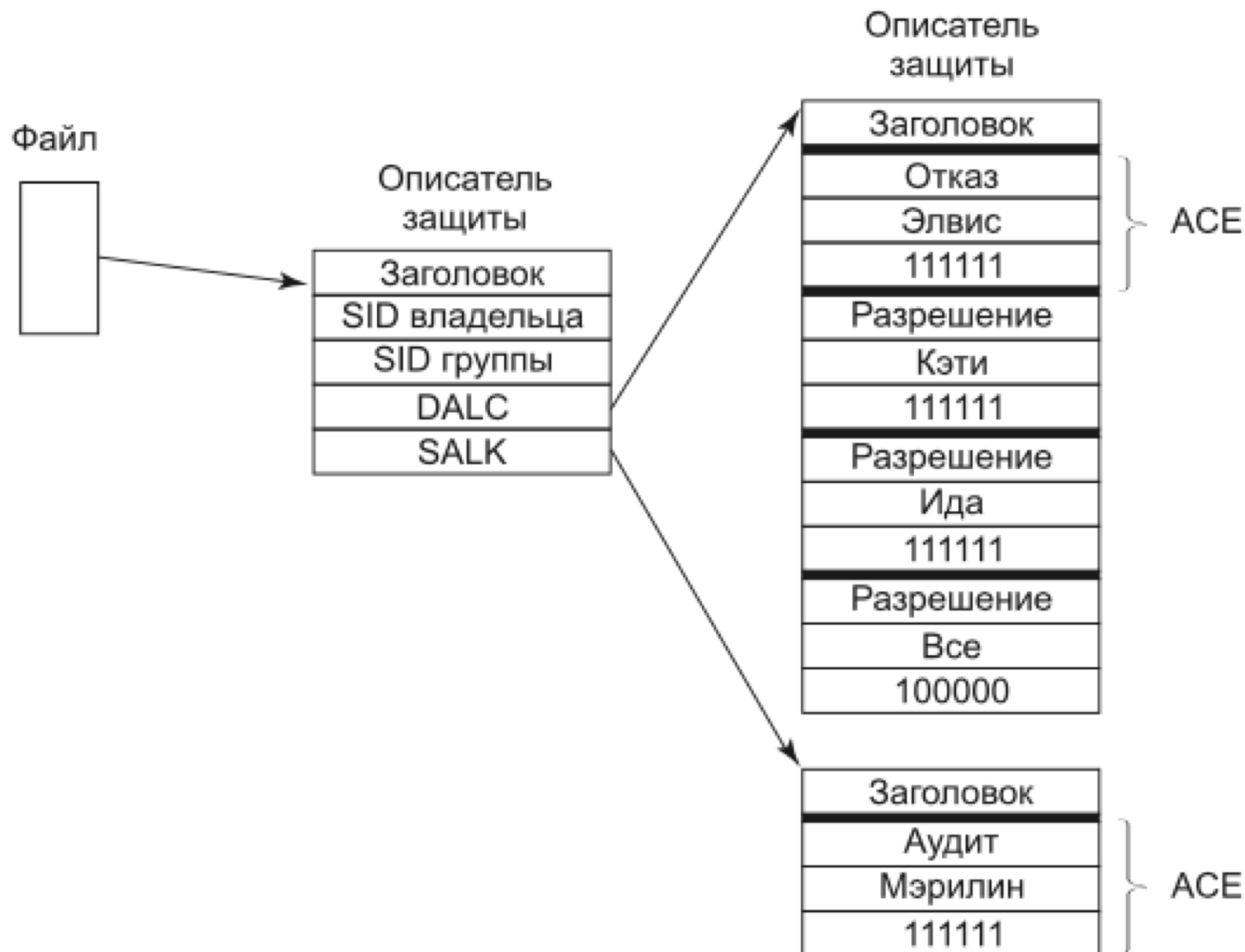
В ОС Windows все типы объектов защищены одинаковым образом. Каждый объект имеет связанный с ним дескриптор безопасности (Security descriptor), который говорит о том, кто и какие операции может выполнять с ним. Дескрипторы безопасности указываются при создании объектов.

Дескриптор безопасности содержит:

- заголовок;
- SID владельца объекта;
- SID первичной группы владельца объекта (для совместимости с POSIX);
- список дискреционного управления доступом DACL (Discretionary ACL), полностью описывающий права различных субъектов на объект;
- список управления системным доступом SACL (System ACL), использующийся для генерации сообщений аудита, связанных с доступом к объекту, содержит также уровень целостности объекта.

Управление доступом в Windows

Дескриптор защиты объекта (дескриптор безопасности)



Управление доступом в Windows

Дескриптор защиты объекта (дескриптор безопасности)

DACL содержит разрешающие и запрещающие доступ списки пользователей и групп, а SACL содержит списки пользователей и групп чьи попытки доступа к данному объекту подлежат аудиту.

Если объект не имеет дескриптора защиты, при обращениях к нему субъектов никакие права доступа не проверяются. В этом случае любой субъект имеет абсолютные права на данный объект.

Если объект имеет дескриптор защиты, но список ACL пустой, то доступ к объекту закрыт для всех субъектов.

Если объект хранится на диске или ином внешнем устройстве, дескриптор защиты хранится вместе с объектом, при этом формат хранения объекта должен предоставлять такую возможность:

- файлы в NTFS имеют дескриптор защиты;
- файлы в других файловых системах не могут иметь дескриптор защиты;
- ключи реестра могут иметь дескриптор защиты независимо от файловой системы диска, на котором размещается реестр.

Управление доступом в Windows

Дескриптор защиты объекта (дескриптор безопасности)

Структура списков DACL и SACL состоит из набора элементов, которые называются *входами управления доступом (Access Control Entry, ACE)* или *элементами управления доступом*. Каждый ACE разрешает или запрещает некоторому субъекту определенные права на доступ к объекту.

В состав ACE могут входить следующие основные поля:

- тип элемента (разрешающий, запрещающий или регистрирующий);
- SID субъекта;
- битовая маска прав доступа;
- флаги, определяющие свойства наследования данного элемента ACE;
- флаги, управляющие аудитом доступа к защищаемому объекту;
- необязательные поля GUID1 и GUID2 (для объектов активного каталога доменов Windows).

Управление доступом в Windows

Проверка прав доступа субъекта к объекту

После формализации атрибутов защиты субъектов и объектов можно рассмотреть схему проверки прав доступа субъектов к объектам.

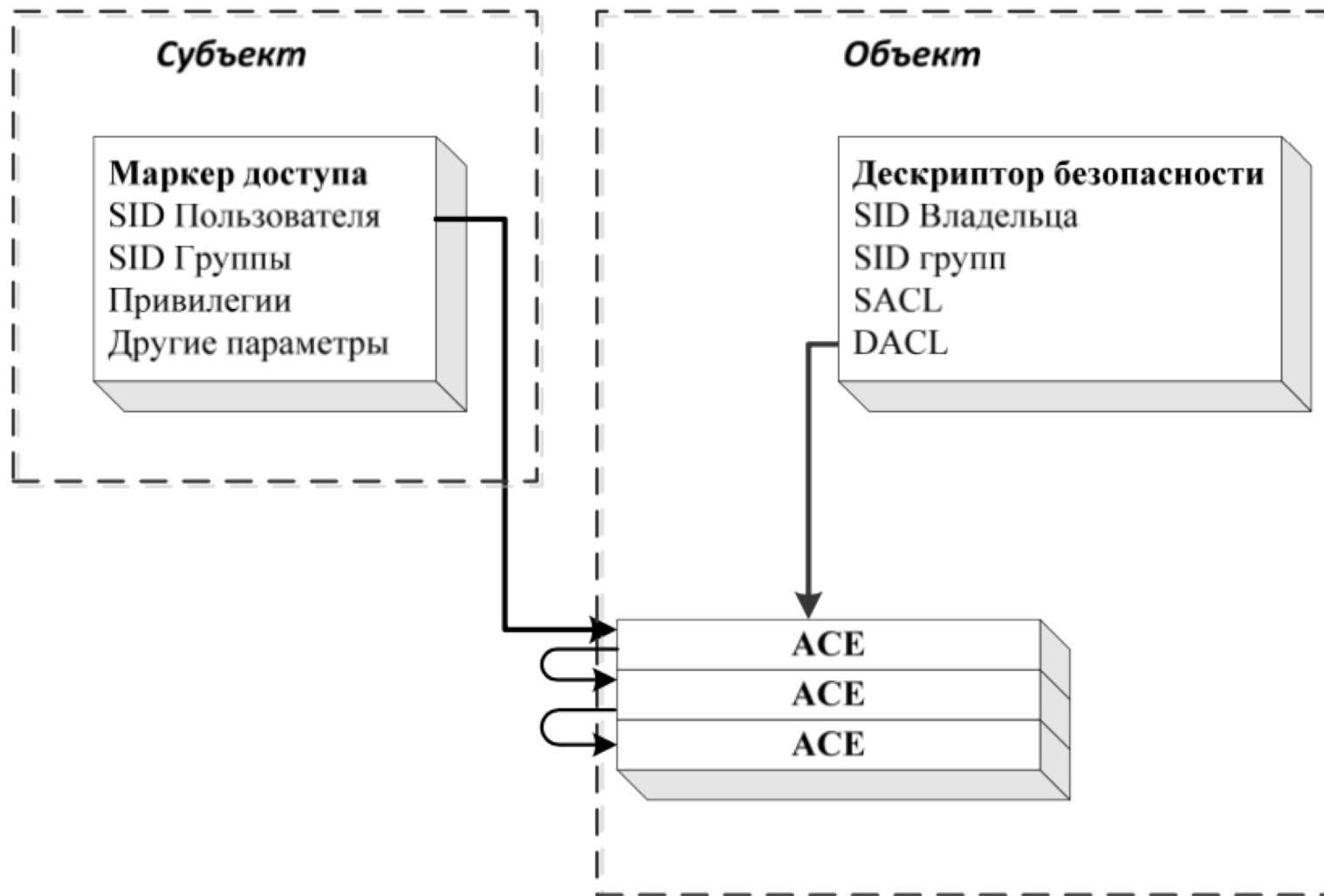
Но для начала стоит напомнить о том, что в ОС Windows в качестве субъекта выступает процесс или поток, который выполняется от имени некоторого пользователя. Доступ к охраняемому объекту выполняется из потока. Поэтому под субъектом часто и понимается исполняемый поток, который при контроле доступа представляется маркером доступа этого потока (первичный или заимствованный у другого процесса при олицетворении).

В свою очередь охраняемый объект, доступ к которому контролируется системой управления безопасностью, представляется при контроле доступа дескриптором безопасности этого объекта.

Управление доступом в Windows

Проверка прав доступа субъекта к объекту

Общая схема проверки прав доступа субъекта к объекту



Управление доступом в Windows

Проверка прав доступа субъекта к объекту

Контроль доступа субъекта к охраняемому объекту выполняется следующим образом.

- При открытии субъектом доступа к охраняемому объекту, что обычно выполняется посредством функций типа Create или Open, система управления безопасностью просматривает список DACL этого охраняемого объекта для поиска элемента ACE, в котором хранится идентификатор безопасности субъекта.
- Если такой элемент в списке DACL не найден, то субъект получает отказ в доступе к объекту.
- Если же такой элемент найден, то система проверяет тип этого элемента (разрешающий или запрещающий).
- Если SID субъекта совпадает с SID владельца объекта и запрашиваются стандартные права доступа, то доступ предоставляется независимо от содержимого DACL.
- Далее система **последовательно** сравнивает SID каждого ACE из DACL с SID маркера доступа.
- Если обнаруживается соответствие, выполняется сравнение маски доступа с проверяемыми правами. Для запрещающих ACE даже при частичном совпадении прав доступ немедленно отклоняется. Для успешной проверки разрешающих элементов необходимо совпадение всех прав.

Управление доступом в Windows

Проверка прав доступа субъекта к объекту

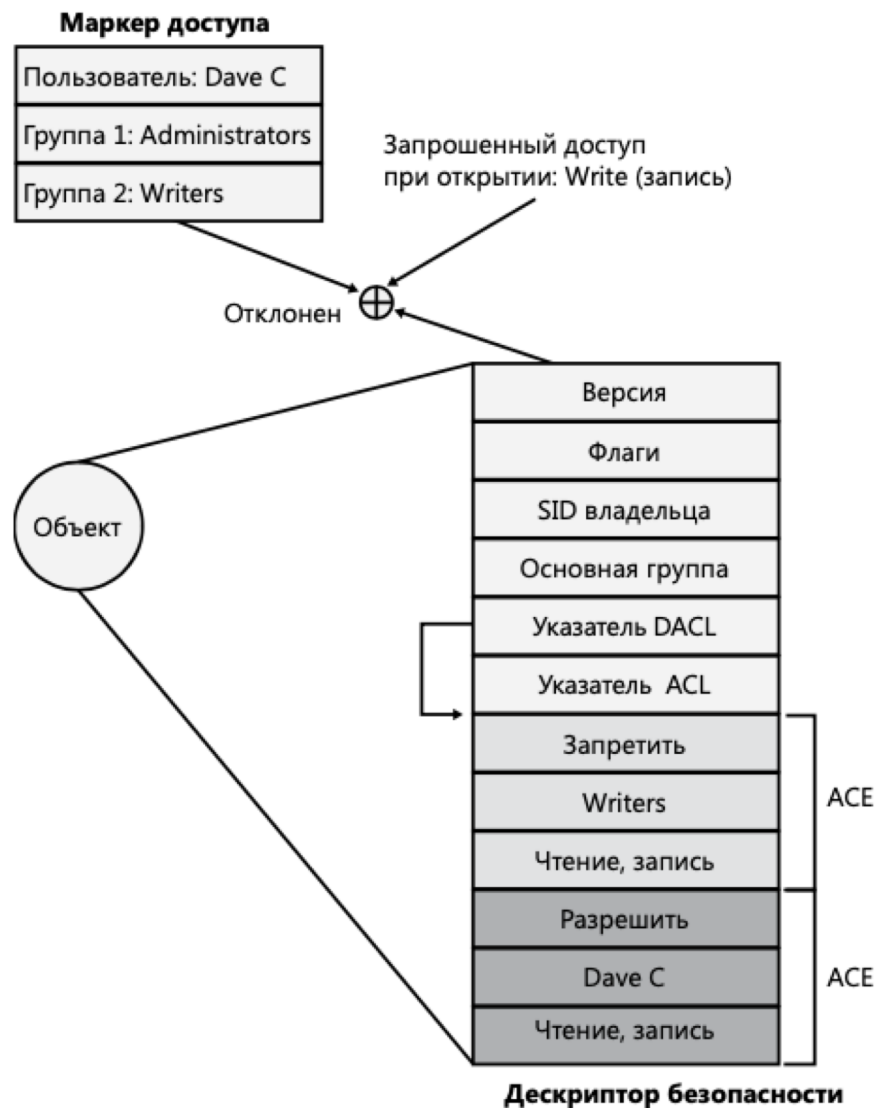
Теоретически может сложиться такая ситуация, когда два ACE противоречат друг другу. Например, один ACE дает полный доступ членам определенной группы, а другой - запрещает доступ определенному пользователю из этой группы. Получит ли этот пользователь доступ к объекту зависит от того, в каком порядке ACE расположены.

Поэтому Microsoft предлагает так называемый *предпочтительный порядок размещения ACE*. Для разрешения противоречий и для ускорения рекомендуется размещать запрещающие элементы перед разрешающими.

После открытия объекта вызывающей стороне возвращается его описатель. При последующих вызовах делается единственная проверка — была ли предпринимаемая сейчас операция в наборе запрошенных в момент открытия операций (чтобы не дать вызывающей стороне открыть файл для чтения, а затем записать в него). Кроме того, вызовы с использованием описателей могут приводить к появлению записей в журналах аудита (если это требуется списком SACL).

Управление доступом в Windows

Проверка прав доступа субъекта к объекту



Управление доступом в Windows

Уровни целостности

В Windows добавлено еще одно средство для решения возникающих при обеспечении безопасности системы (при помощи ACL) проблем.

Это новый обязательный *идентификатор уровня целостности (Integrity-level SID)* в маркере доступа процесса, причем объекты указывают список ACE уровня целостности в списке SACL. Уровень целостности предотвращает доступ для записи к объектам (вне зависимости от того, какие ACE есть в DACL).

В частности, схема уровня целостности используется для защиты от скомпрометированного процесса Internet Explorer, который, возможно, был атакован из-за скачивания кода с незнакомого веб-сайта. Internet Explorer с низкими правами (он называется *Low-rights IE*) работает с установленным в значение low уровнем целостности. По умолчанию все файлы и ключи реестра имеют уровень целостности medium, так что работающий с уровнем low процесс Internet Explorer не может их модифицировать.

Управление доступом в Windows

Назначение дескрипторов защиты создаваемым объектам

При создании в ОС Windows нового объекта ему назначаются атрибуты защиты согласно следующим правилам.

- Если процесс, создающий объект, явно указывает корректный дескриптор защиты для создаваемого объекта, создаваемому объекту назначается указанный дескриптор защиты.
- Если процесс, создающий объект, указывает, что атрибуты защиты должны быть установлены по умолчанию, или если указанный дескриптор защиты некорректен, дескриптор защиты объекта создается с помощью механизма наследования.
- Если по каким-то причинам наследование дескриптора защиты невозможно, объекту присваивается дескриптор защиты на основе данных, хранящихся в маркере доступа субъекта, создающего объект.

Управление доступом в Windows

Назначение дескрипторов защиты создаваемым объектам

- DACL создаваемого дескриптора защиты формируется при наследовании из ACE, входящих в DACL объекта-родителя. То, какие ACE объекта-родителя будут включены в DACL создаваемого объекта, определяется следующими флагами ACE:
 - CONTAINER_INHERIT_ACE – если этот флаг установлен и создаваемый объект является контейнером, данный ACE должен включаться в DACL создаваемого объекта;
 - OBJECT_INHERIT_ACE – если этот флаг установлен и создаваемый объект не является контейнером, данный ACE должен включаться в DACL создаваемого объекта;
 - NO_PROPAGATE_INHERIT_ACE – если этот флаг установлен, при наследовании ACE флаги CONTAINER_INHERIT_ACE и OBJECT_INHERIT_ACE сбрасываются. Другими словами, при наличии этого флага ACE наследуется только один раз;
 - INHERIT_ONLY_ACE – если этот флаг установлен, данный ACE игнорируется при проверке прав доступа к объекту и используется только при наследовании.

Управление доступом в Windows

Назначение дескрипторов защиты создаваемым объектам

Детали использования флагов наследования для создаваемых объектов, являющихся и не являющихся контейнерами, см. в [Проскурин В.Г., 2014].

SACL создаваемого объекта наследуется по тем же правилам, что и DACL.

Начиная с версии Windows 2000, поддерживается функция автоматического наследования изменений в DACL и SACL при каждом изменении в дескрипторе защиты контейнера (управляется соответствующими флагами дескриптора защиты дочернего объекта). Контейнеры при этом обходятся рекурсивно.

Унаследованные ACE всегда располагаются после явно назначенных и, следовательно, имеют более низкий приоритет.



Управление доступом в Windows

Мандатный контроль целостности (MIC & UAC)

Элементы изолированной среды (интерфейс SAFER)